

What is claimed is:

1 1. A client-server system capable of validating cached data comprising:
2 a data store for storing data;
3 a server for retrieving and updating data in the data store to service client
4 requests;
5 a transformation engine for transforming data into a format suitable for a
6 client application based on a set of transformation rules;
7 a cache for temporarily storing transformed data as data objects for later
8 reuse;
9 a cache monitor for ensuring that cached objects are validated when
10 changes to data in the data store are detected by the server; and
11 an object dependency mapper for automatically and continuously
12 determining dependencies between data in the data store and sets of
13 transformation rules.

1 2. The system as recited in claim 1 further comprising an object manager
2 for managing data objects in the cache.

1 3. The system as recited in claim 2 further comprising a transformation
2 rule alert service for detecting when the transformation rules are modified, added to
1 the system and deleted from the system.

1 4. The system as recited in claim 3, wherein the server accesses the
2 object manager to generate a response to a client request for data.

1 5. The system as recited in claim 4, wherein the server accesses the
2 cache monitor to validate cached objects when a data update request is received.

1 6. The system as recited in claim 5, wherein:
2 data in the data store is represented as a tree structure having a root node, a
3 plurality of intermediate nodes and leaf nodes, the leaf nodes representing data in
4 the data store; and
5 a transformation rule is an expression describing a path from the root node
6 to a particular node in the tree.

1 7. The system as recited in claim 6, wherein:
2 a set of the transformation rules constitutes a style sheet; and
3 the transformation engine receives a style sheet and the data tree as input,
4 and outputs a transformed data object.

1 8. The system as recited in claim 7, wherein the cache includes a
2 plurality of data objects each associated with a style sheet used to generate said
3 each object.

1 9. The system as recited in claim 8, wherein:
2 the object manager uses the transformation engine to generate a new object
3 in response to a client request when the new object does not exist in the cache;
4 and
5 the object manager stores the new object in the cache automatically.

1 10. The system as recited in claim 9, wherein the object manager
2 periodically refreshes the cache and removes the objects that have been flagged as
3 invalid by the cache monitor.

1 11. The system as recited in claim 9, wherein the object manager
2 optionally maintains statistical information for each object in the cache, and
3 automatically removes cached objects that are being accessed infrequently by the
4 clients.

1 12. The system as recited in claim 7, wherein the object dependency
2 mapper includes a table of dependencies, each dependency associating a
3 transformation rule with the style sheets that include the transformation rule.

1 13. The system as recited in claim 12, wherein the table of dependencies
2 is automatically generated and maintained by the object dependency mapper.

1 14. The system as recited in claim 12, wherein the transformation rule
2 alert service communicates updates on the style sheets to the object dependency
3 mapper.

1 15. The system as recited in claim 12, wherein the cache monitor uses
2 the table of dependencies to determine a set of relevant style sheets, said relevant
3 style sheets referencing a node in the data tree related to a data update; and
4 the cache monitor accesses the cache to invalidate the objects generated by
5 the relevant style sheets.

1 16. In a client-sever computing system having a cache and storing data
2 as data objects, a method for determining invalid cached objects comprising the
3 steps of:

4 transforming data into a format suitable for a client application based on a
5 set of transformation rules;

6 determining dependencies between cached objects and data related to the
7 cached objects;

8 monitoring updates to the related data; and

9 determining the cached objects that are affected by changes to the related
10 data based on the dependencies.

1 17. The method as recited in claim 16, wherein the transformed format is
2 html.

1 18. The method as recited in claim 16, wherein:
2 a set of transformation rules constitutes a style sheet;
3 each dependency associates a transformation rule with a style sheet; and
4 each data object is the data transformed by a style sheet.

1 19. The method as recited in claim 16, wherein:
2 data is represented as a tree structure having a plurality of nodes; and
3 the cached objects that are affected by the data changes are determined
4 using the tree structure.

1 20. The method as recited in claim 19, wherein the dependencies are
2 maintained in a table of dependencies.

1 21. The method as recited in claim 20, wherein the step of determining
2 the affected objects comprises the steps of:

3 identifying the nodes associated with data updates;
4 identifying the transformation rules related to the identified nodes;
5 determining a set of relevant style sheets using the table of dependencies,
6 the relevant style sheets including the identified transformation rules; and
7 identifying the cached objects that have been transformed by the relevant
8 style sheets.

1 22. In a client-sever computing system having a cache and storing data
2 as data objects, a computer-program product for determining invalid cached objects
3 comprising:

4 a computer-readable medium;
5 means, provided on the computer-readable medium, for transforming data
6 into a format suitable for a client application based on a set of transformation rules;
7 means, provided on the computer-readable medium, for determining
8 dependencies between cached objects and data related to the cached objects; and
9 means, provided on the computer-readable medium, for monitoring updates
10 to the related data;
11 means, provided on the computer-readable medium, for determining the
12 cached objects that are affected by changes to the related data based on the
13 dependencies.

1 23. The computer-program product as recited in claim 22, wherein the
2 transformed format is html.

1 24. The computer-program product as recited in claim 22, wherein:
2 a set of transformation rules constitutes a style sheet;
3 each dependency associates a transformation rule with a style sheet; and
4 each data object is the data transformed by a style sheet.

1 25. The computer-program product as recited in claim 22, wherein:
2 data is represented as a tree structure having a plurality of nodes; and
3 the cached objects that are affected by the data changes are determined
4 using the tree structure.

1 26. The method as recited in claim 25, wherein the dependencies are
2 maintained in a table of dependencies.

1 27. The computer-program product as recited in claim 26, wherein the
2 means for determining the affected cached objects comprises:
3 means, provided on the computer-readable medium, for identifying the
4 nodes associated with data updates;
5 means, provided on the computer-readable medium, for identifying the
6 transformation rules related to the identified nodes;
7 means, provided on the computer-readable medium, for determining a set of
8 relevant style sheets using the table of dependencies, the relevant style sheets
9 including the identified transformation rules; and
10 means, provided on the computer-readable medium, for identifying the
11 cached objects that have been transformed by the relevant style sheets.